# Chapter 3
# Nonlinear Model Predictive Control

In this chapter, we introduce the nonlinear model predictive control algorithm in a rigorous way. We start by defining a basic NMPC algorithm for constant reference and continue by formalizing state and control constraints. Viability (or weak forward invariance) of the set of state constraints is introduced and the consequences for the admissibility of the NMPC feedback law are discussed. After having introduced NMPC in a special setting, we describe various extensions of the basic algorithm, considering time varying reference solutions, terminal constraints and costs and additional weights. Finally, we investigate the optimal control problem corresponding to this generalized setting and prove several properties, most notably the dynamic programming principle.

## 3.1 The Basic NMPC Algorithm

As already outlined in the introductory Chap. 1, the idea of the NMPC scheme is as follows: at each sampling instant $n$ we optimize the predicted future behavior of the system over a finite time horizon $k = 0, \ldots, N-1$ of length $N \geq 2$ and use the first element of the resulting optimal control sequence as a feedback control value for the next sampling interval. In this section we give a detailed mathematical description of this basic idea for a constant reference $x^{\mathrm{ref}} \equiv x_* \in X$. The time varying case as well as several other variants will then be presented in Sect. 3.3.

A prerequisite for being able to find a feedback law which stabilizes the system at $x_*$ is that $x_*$ is an equilibrium of the nominal closed-loop system (2.5), i.e., $x_* = f(x_*, \mu(x_*))$—this follows immediately from Definition 2.14 with $g(x) = f(x, \mu(x))$. A necessary condition for this is that there exists a control value $u_* \in U$ with

$$x_* = f(x_*, u_*), \tag{3.1}$$

which we will assume in the sequel. The cost function to be used in our optimization should penalize the distance of an arbitrary state $x \in X$ to $x_*$. In addition, it is often

L. Grüne, J. Pannek, *Nonlinear Model Predictive Control*,
Communications and Control Engineering,
DOI 10.1007/978-0-85729-501-9_3, © Springer-Verlag London Limited 2011

desired to penalize the control $u \in U$. This can be useful for computational reasons, because optimal control problems may be easier to solve if the control variable is penalized. On the other hand, penalizing $u$ may also be desired for modeling purposes, e.g., because we want to avoid the use of control values $u \in U$ corresponding to expensive high energy. For these reasons, we choose our cost function to be of the form $\ell : X \times U \to \mathbb{R}_0^+$.

In any case, we require that if we are in the equilibrium $x_*$ and use the control value $u_*$ in order to stay in the equilibrium, then the cost should be 0. Outside the equilibrium, however, the cost should be positive, i.e.,

$$\ell(x_*, u_*) = 0 \quad \text{and} \quad \ell(x, u) > 0 \quad \text{for all } x \in X,\ u \in U \text{ with } x \neq x_*. \tag{3.2}$$

If our system is defined on Euclidean space, i.e., $X = \mathbb{R}^d$ and $U = \mathbb{R}^m$, then we may always assume $x_* = 0$ and $u_* = 0$ without loss of generality: if this is not the case we can replace $f(x, u)$ by $f(x + x_*, u + u_*) - x_*$ which corresponds to a simple linear coordinate transformation on $X$ and $U$. Indeed, this transformation is always possible if $X$ and $U$ are vector spaces, even if they are not Euclidean spaces. In this case, a popular choice for $\ell$ meeting condition (3.2) is the quadratic function

$$\ell(x, u) = \|x\|^2 + \lambda \|u\|^2,$$

with the usual Euclidean norms and a parameter $\lambda \geq 0$. In our general setting on metric spaces with metrics $d_X$ and $d_U$ on $X$ and $U$, the analogous choice of $\ell$ is

$$\ell(x, u) = d_X(x, x_*)^2 + \lambda d_U(u, u_*)^2. \tag{3.3}$$

Note, however, that in both settings many other choices are possible and often reasonable, as we will see in the subsequent chapters. Moreover, we will introduce additional conditions on $\ell$ later, which we require for a rigorous stability proof of the NMPC closed loop.

In the case of sampled data systems we can take the continuous time nature of the underlying model into account by defining $\ell$ as an integral over a continuous time cost function $L : X \times U \to \mathbb{R}_0^+$ on a sampling interval. Using the continuous time solution $\varphi$ from (2.8), we can define

$$\ell(x, u) := \int_0^T L\big(\varphi(t, 0, x, u), u(t)\big)\, dt. \tag{3.4}$$

Defining $\ell$ this way, we can incorporate the intersampling behavior of the sampled data system explicitly into our optimal control problem. As we will see later in Remark 4.13, this enables us to derive rigorous stability properties not only for the sampled data closed-loop system (2.30). The numerical computation of the integral in (3.4) can be efficiently integrated into the numerical solution of the ordinary differential equation (2.6), see Sect. 9.4 for details.

Given such a cost function $\ell$ and a prediction horizon length $N \geq 2$, we can now formulate the basic NMPC scheme as an algorithm. In the optimal control problem ($\text{OCP}_\text{N}$) within this algorithm we introduce a set of control sequences $\mathbb{U}^N(x_0) \subseteq U^N$ over which we optimize. This set may include constraints depending on the initial value $x_0$. Details about how this set should be chosen will be discussed in Sect. 3.2. For the moment we simply set $\mathbb{U}^N(x_0) := U^N$ for all $x_0 \in X$.

**Algorithm 3.1** (Basic NMPC algorithm for constant reference $x^{\mathrm{ref}} \equiv x_*$) At each sampling time $t_n$, $n = 0, 1, 2 \ldots$:

(1) Measure the state $x(n) \in X$ of the system.
(2) Set $x_0 := x(n)$, solve the optimal control problem

$$\boxed{\begin{aligned} &\text{minimize} \quad J_N\big(x_0, u(\cdot)\big) := \sum_{k=0}^{N-1} \ell\big(x_u(k, x_0), u(k)\big) \\ &\text{with respect to} \quad u(\cdot) \in \mathbb{U}^N(x_0), \quad \text{subject to} \\ &x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big) \end{aligned}} \tag{$\mathrm{OCP_N}$}$$

and denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N(x_0)$.
(3) Define the NMPC-feedback value $\mu_N(x(n)) := u^\star(0) \in U$ and use this control value in the next sampling period.

Observe that in this algorithm we have assumed that an optimal control sequence $u^\star(\cdot)$ exists. Sufficient conditions for this existence are briefly discussed after Definition 3.14, below.

The nominal closed-loop system resulting from Algorithm 3.1 is given by (2.5) with state feedback law $\mu = \mu_N$, i.e.,

$$x^+ = f\big(x, \mu_N(x)\big). \tag{3.5}$$

The trajectories of this system will be denoted by $x_{\mu_N}(n)$ or, if we want to emphasize the initial value $x_0 = x_{\mu_N}(0)$, by $x_{\mu_N}(n, x_0)$.

During our theoretical investigations we will neglect the fact that computing the solution of ($\mathrm{OCP_N}$) in Step (2) of the algorithm usually needs some computation time $\tau_c$ which—in the case when $\tau_c$ is relatively large compared to the sampling period $T$—may not be negligible in a real time implementation. We will sketch a solution to this problem in Sect. 7.6.

In our abstract formulations of the NMPC Algorithm 3.1 only the first element $u^\star(0)$ of the respective minimizing control sequence is used in each step, the remaining entries $u^\star(1), \ldots, u^\star(N-1)$ are discarded. In the practical implementation, however, these entries play an important role because numerical optimization algorithms for solving ($\mathrm{OCP_N}$) (or its variants) usually work iteratively: starting from an initial guess $u^0(\cdot)$ an optimization algorithm computes iterates $u^i(\cdot)$, $i = 1, 2, \ldots$ converging to the minimizer $u^\star(\cdot)$ and a good choice of $u^0(\cdot)$ is crucial in order to obtain fast convergence of this iteration, or even to ensure convergence, at all. Here, the minimizing sequence from the previous time step can be efficiently used in order to construct such a good initial guess. Several different ways to implement this idea are discussed in Sect. 10.4.

## 3.2 Constraints

One of the main reasons for the success of NMPC (and MPC in general) is its ability to explicitly take constraints into account. Here, we consider constraints both on

the control as well as on the state. To this end, we introduce a nonempty *state constraint set* $\mathbb{X} \subseteq X$ and for each $x \in \mathbb{X}$ we introduce a nonempty *control constraint set* $\mathbb{U}(x) \subseteq U$. Of course, $\mathbb{U}$ may also be chosen independent of $x$. The idea behind introducing these sets is that we want the trajectories to lie in $\mathbb{X}$ and the corresponding control values to lie in $\mathbb{U}(x)$. This is made precise in the following definition.

**Definition 3.2** Consider a control system (2.1) and the state and control constraint sets $\mathbb{X} \subseteq X$ and $\mathbb{U}(x) \subseteq U$.

(i) The states $x \in \mathbb{X}$ are called *admissible states* and the control values $u \in \mathbb{U}(x)$ are called *admissible control values for* $x$.

(ii) For $N \in \mathbb{N}$ and an initial value $x_0 \in \mathbb{X}$ we call a control sequence $u \in U^N$ and the corresponding trajectory $x_u(k, x_0)$ *admissible for* $x_0$ *up to time* $N$, if

$$u(k) \in \mathbb{U}\big(x_u(k, x_0)\big) \quad \text{and} \quad x_u(k+1, x_0) \in \mathbb{X}$$

hold for all $k = 0, \ldots, N-1$. We denote the set of admissible control sequences for $x_0$ up to time $N$ by $\mathbb{U}^N(x_0)$.

(iii) A control sequence $u \in U^\infty$ and the corresponding trajectory $x_u(k, x_0)$ are called *admissible for* $x_0$ if they are admissible for $x_0$ up to every time $N \in \mathbb{N}$. We denote the set of admissible control sequences for $x_0$ by $\mathbb{U}^\infty(x_0)$.

(iv) A (possibly time varying) feedback law $\mu : \mathbb{N}_0 \times X \to U$ is called *admissible* if $\mu(n, x) \in \mathbb{U}^1(x)$ holds for all $x \in \mathbb{X}$ and all $n \in \mathbb{N}_0$.

Whenever the reference to $x$ or $x_0$ is clear from the context we will omit the additional "for $x$" or "for $x_0$".

Since we can (and will) identify control sequences with only one element with the respective control value, we can consider $\mathbb{U}^1(x_0)$ as a subset of $U$, which we already implicitly did in the definition of admissibility for the feedback law $\mu$, above. However, in general $\mathbb{U}^1(x_0)$ does not coincide with $\mathbb{U}(x_0) \subseteq U$ because using $x_u(1, x) = f(x, u)$ and the definition of $\mathbb{U}^N(x_0)$ we get $\mathbb{U}^1(x) := \{u \in \mathbb{U}(x) \mid f(x, u) \in \mathbb{X}\}$. With this subtle difference in mind, one sees that our admissibility condition (iv) on $\mu$ ensures both $\mu(n, x) \in \mathbb{U}(x)$ and $f(x, \mu(n, x)) \in \mathbb{X}$ whenever $x \in \mathbb{X}$.

Furthermore, our definition of $\mathbb{U}^N(x)$ implies that even if $\mathbb{U}(x) = \mathbb{U}$ is independent of $x$ the set $\mathbb{U}^N(x)$ may depend on $x$ for some or all $N \in \mathbb{N}_\infty$.

Often, in order to be suitable for optimization purposes these sets are assumed to be compact and convex. For our theoretical investigations, however, we do not need any regularity requirements of this type except that these sets are nonempty. We will, however, frequently use the following assumption.

**Assumption 3.3** For each $x \in \mathbb{X}$ there exists $u \in \mathbb{U}(x)$ such that $f(x, u) \in \mathbb{X}$ holds.

The property defined in this assumption is called *viability* or *weak* (*or controlled*) *forward invariance* of $\mathbb{X}$. It excludes the situation that there are states $x \in \mathbb{X}$ from which the trajectory leaves the set $\mathbb{X}$ for all admissible control values. Hence, it ensures $\mathbb{U}^N(x_0) \neq \emptyset$ for all $x_0 \in \mathbb{X}$ and all $N \in \mathbb{N}_\infty$. This property is

important to ensure the feasibility of ($\mathrm{OCP_N}$): the optimal control problem ($\mathrm{OCP_N}$) is called *feasible* for an initial value $x_0$ if the set $\mathbb{U}^N(x_0)$ over which we optimize is nonempty. Viability of $\mathbb{X}$ thus implies that ($\mathrm{OCP_N}$) is feasible for each $x_0 \in \mathbb{X}$ and hence ensures that $\mu_N(x)$ is well defined for each $x \in \mathbb{X}$. Furthermore, a straightforward induction shows that under Assumption 3.3 any finite admissible control sequence $u(\cdot) \in \mathbb{U}^N(x_0)$ can be extended to an infinite admissible control sequence $\tilde{u}(\cdot) \in \mathbb{U}^\infty(x_0)$ with $u(k) = \tilde{u}(k)$ for all $k = 0, \ldots, N-1$.

In order to see that the construction of a constraint set $\mathbb{X}$ meeting Assumption 3.3 is usually a nontrivial task, we reconsider Example 2.2.

*Example 3.4* Consider Example 2.2, i.e.,

$$x^+ = f(x,u) = \begin{pmatrix} x_1 + x_2 + u/2 \\ x_2 + u \end{pmatrix}.$$

Assume we want to constrain all variables, i.e., the position $x_1$, the velocity $x_2$ and the acceleration $u$ to the interval $[-1,1]$. For this purpose one could define $\mathbb{X} = [-1,1]^2$ and $\mathbb{U}(x) = \mathbb{U} = [-1,1]$. Then, however, for $x = (1,1)^\top$, one immediately obtains

$$x_1^+ = x_1 + x_2 + u/2 = 2 + u/2 \geq 3/2$$

for all $u$, hence $x^+ \notin \mathbb{X}$ for all $u \in \mathbb{U}$. Thus, in order to find a viable set $\mathbb{X}$ we need to either tighten or relax some of the constraints. For instance, relaxing the constraint on $u$ to $\mathbb{U} = [-2,2]$ the viability of $\mathbb{X} = [-1,1]^2$ is guaranteed, because then by elementary computations one sees that for each $x \in \mathbb{X}$ the control value

$$u = \begin{cases} 0, & x_1 + x_2 \in [-1,1], \\ 2 - 2x_1 - 2x_2, & x_1 + x_2 > 1, \\ -2 - 2x_1 - 2x_2, & x_1 + x_2 < -1 \end{cases}$$

is in $\mathbb{U}$ and satisfies $f(x,u) \in \mathbb{X}$. A way to achieve viability without changing $\mathbb{U}$ is by tightening the constraint on $x_2$ by defining

$$\mathbb{X} = \left\{ (x_1, x_2)^T \in \mathbb{R}^2 \,\middle|\, x_1 \in [-1,1],\ x_2 \in [-1,1] \cap [-3/2 - x_1, 3/2 - x_1] \right\}, \tag{3.6}$$

see Fig. 3.1. Again, elementary computations show that for each $x \in \mathbb{X}$ and

$$u = \begin{cases} 1, & x_2 < -1/2, \\ -2x_2, & x_2 \in [-1/2, 1/2], \\ -1, & x_2 > 1/2 \end{cases}$$

the desired properties $u \in \mathbb{U}$ and $f(x,u) \in \mathbb{X}$ hold.

This example shows that finding viable constraint sets $\mathbb{X}$ (and the corresponding $\mathbb{U}$ or $\mathbb{U}(x)$) is a tricky task already for very simple systems. Still, Assumption 3.3 significantly simplifies the subsequent analysis, cf. Theorem 3.5, below. For this reason we will impose this condition in our theoretical investigations for schemes

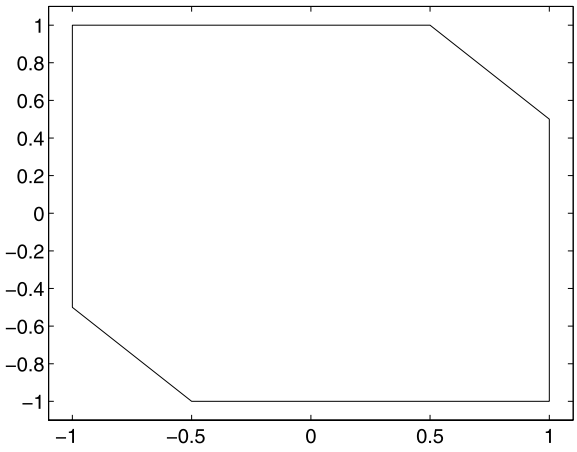


**Fig. 3.1** Illustration of the set $\mathbb{X}$ from (3.6)

without stabilizing terminal constraints in Chap. 6. Ways to relax this condition will be discussed in Sects. 8.1–8.3.

For schemes with stabilizing terminal constraints as featured in Chap. 5 we will not need this assumption, since for these schemes the region on which the NMPC controller is defined is by construction confined to feasible subsets $\mathbb{X}_N$ of $\mathbb{X}$, see Definition 3.9, below. Even if $\mathbb{X}$ is not viable, these feasible sets $\mathbb{X}_N$ turn out to be viable provided the terminal constraint set is viable, cf. Lemmas 5.2 and 5.10. For a more detailed discussion of these issues see also Part (iv) of the discussion in Sect. 8.4.

NMPC is well suited to handle constraints because these can directly be inserted into Algorithm 3.1. In fact, since we already formulated the corresponding optimization problem ($\mathrm{OCP_N}$) with state dependent control value sets, the constraints are readily included if we use $\mathbb{U}^N(x_0)$ from Definition 3.2(ii) in ($\mathrm{OCP_N}$). The following theorem shows that the viability assumption ensures that the NMPC closed-loop system obtained this way indeed satisfies the desired constraints.

**Theorem 3.5** *Consider Algorithm 3.1 using $\mathbb{U}^N(x_0)$ from Definition 3.2(ii) in the optimal control problem ($\mathrm{OCP_N}$) for constraint sets $\mathbb{X} \subset X$, $\mathbb{U}(x) \subset U$, $x \in \mathbb{X}$, satisfying Assumption 3.3. Consider the nominal closed-loop system (3.5) and suppose that $x_{\mu_N}(0) \in \mathbb{X}$. Then the constraints are satisfied along the solution of (3.5), i.e.,*

$$x_{\mu_N}(n) \in \mathbb{X} \quad \text{and} \quad \mu_N\big(x_{\mu_N}(n)\big) \in \mathbb{U}\big(x_{\mu_N}(n)\big) \tag{3.7}$$

*for all $n \in \mathbb{N}$. Thus, the NMPC-feedback $\mu_N$ is admissible in the sense of Definition 3.2(iv).*

*Proof* First, recall from the discussion after Assumption 3.3 that under this assumption the optimal control problem ($\mathrm{OCP_N}$) is feasible for each $x \in \mathbb{X}$, hence $\mu_N(x)$ is well defined for each $x \in \mathbb{X}$.

We now show that $x_{\mu_N}(n) \in \mathbb{X}$ implies $\mu_N(x_{\mu_N}(n)) \in \mathbb{U}(x_{\mu_N}(n))$ and $x_{\mu_N}(n+1) \in \mathbb{X}$. Then the assertion follows by induction from $x_{\mu_n}(0) \in \mathbb{X}$.

The viability of $\mathbb{X}$ from Assumption 3.3 ensures that whenever $x_{\mu_N}(n) \in \mathbb{X}$ holds in Algorithm 3.1, then $x_0 \in \mathbb{X}$ holds for the respective optimal control problem ($\mathrm{OCP_N}$). Since the optimization is performed with respect to admissible control sequences only, also the optimal control sequence $u^\star(\cdot)$ is admissible for $x_0 = x_{\mu_N}(n)$. This implies $\mu_N(x_{\mu_N}(n)) = u^\star(0) \in \mathbb{U}^1(x_{\mu_N}(n)) \subseteq \mathbb{U}(x_{\mu_N}(n))$ and thus also

$$x_{\mu_N}(n+1) = f\big(x_{\mu_N}(n), \mu_N\big(x_{\mu_N}(n)\big)\big) = f\big(x(n), u^\star(0)\big) \in \mathbb{X},$$

i.e., $x_{\mu_N}(n+1) \in \mathbb{X}$. □

Theorem 3.5 in particular implies that if a state $x$ is feasible for ($\mathrm{OCP_N}$), which under Assumption 3.3 is equivalent to $x \in \mathbb{X}$ (cf. the discussion after Assumption 3.3), then its closed-loop successor state $f(x, \mu_N(x))$ is again feasible. This property is called *recursive feasibility* of $\mathbb{X}$.

In the case of sampled data systems, the constraints are only defined for the sampling times $t_n$ but not for the intersampling times $t \neq t_n$. That is, for the sampled data closed-loop system (2.30) we can only guarantee

$$\varphi(t_n, t_0, x_0, \mu) \in \mathbb{X} \quad \text{for } n = 0, 1, 2, \ldots$$

but in general not

$$\varphi(t, t_0, x_0, \mu) \in \mathbb{X} \quad \text{for } t \neq t_n,\ n = 0, 1, 2, \ldots.$$

Since we prefer to work within the discrete time framework, directly checking $\varphi(t, t_0, x_0, u) \in \mathbb{X}$ for all $t$ does not fit our setting. If desired, however, one could implicitly include this condition in the definition of $\mathbb{U}(x)$, e.g., by defining new control constraint sets via

$$\widetilde{\mathbb{U}}(x) := \big\{u \in \mathbb{U}(x) \;\big|\; \varphi(t, 0, x, u) \in \mathbb{X} \text{ for all } t \in [0, T]\big\}.$$

In practice, however, this is often not necessary because continuity of $\varphi$ in $t$ ensures that the constraints are usually only "mildly" violated for $t \neq t_n$, i.e., $\varphi(t, t_0, x_0, \mu)$ will still be close to $\mathbb{X}$ at intersampling times. Still, one should keep this fact in mind when designing the constraint set $\mathbb{X}$.

In the underlying optimization algorithms for solving ($\mathrm{OCP_N}$), usually the constraints cannot be specified via sets $\mathbb{X}$ and $\mathbb{U}(x)$. Rather, one uses so-called *equality* and *inequality constraints* in order to specify $\mathbb{X}$ and $\mathbb{U}(x)$ according to the following definition.

**Definition 3.6** Given functions $G_i^S : X \times U \to \mathbb{R}$, $i \in \mathcal{E}^S = \{1, \ldots, p_g\}$ and $H_i^S : X \times U \to \mathbb{R}$, $i \in \mathcal{I}^S = \{p_g + 1, \ldots, p_g + p_h\}$ with $r_g, r_h \in \mathbb{N}_0$, we define the constraint sets $\mathbb{X}$ and $\mathbb{U}(x)$ via

$$\begin{aligned}\mathbb{X} := \big\{x \in X \;\big|\; &\text{there exists } u \in U \text{ with } G_i^S(x, u) = 0 \text{ for all } i \in \mathcal{E}^S \\ &\text{and } H_i^S(x, u) \geq 0 \text{ for all } i \in \mathcal{I}^S\big\}\end{aligned}$$

and, for $x \in \mathbb{X}$

$$\mathbb{U}(x) := \big\{u \in U \;\big|\; G_i^S(x,u) = 0 \text{ for all } i \in \mathcal{E}^S \text{ and } H_i^S(x,u) \geq 0 \text{ for all } i \in \mathcal{I}^S\big\}.$$

Here, the functions $G_i^S$ and $H_i^S$ do not need to depend on both arguments. The functions $G_i^S$, $H_i^S$ not depending on $u$ are called *pure state constraints*, the functions $G_i^S$, $H_i^S$ not depending on $x$ are called *pure control constraints* and the functions $G_i^S$, $H_i^S$ depending on both $x$ and $u$ are called *mixed constraints*.

Observe that if we do not have mixed constraints then $\mathbb{U}(x)$ is independent of $x$.

The reason for defining $\mathbb{X}$ and $\mathbb{U}(x)$ via these (in)equality constraints is purely algorithmic: the plain information "$x_u(k,x_0) \notin \mathbb{X}$" does not yield any information for the optimization algorithm in order to figure out how to find an admissible $u(\cdot)$, i.e., a $u(\cdot)$ for which "$x_u(k,x_0) \in \mathbb{X}$" holds. In contrast to that, an information of the form "$H_i^S(x_u(k,x_0),u(k)) < 0$" together with additional knowledge about $H_i^S$ (provided, e.g., by the derivative of $H_i^S$) enables the algorithm to compute a "direction" in which $u(\cdot)$ needs to be modified in order to reach an admissible $u(\cdot)$. For more details on this we refer to Chap. 10.

In our theoretical investigations we will use the notationally more convenient set characterization of the constraints via $\mathbb{X}$ and $\mathbb{U}(x)$ or $\mathbb{U}^N(x)$. In the practical implementation of our NMPC method, however, we will use their characterization via the inequality constraints from Definition 3.6.

## 3.3 Variants of the Basic NMPC Algorithms

In this section we discuss some important variants and extensions of the basic NMPC Algorithm 3.1; several further variants will be briefly discussed in Sect. 3.5. We start by incorporating non-constants references $x^{\mathrm{ref}}(n)$ and afterwards turn to including terminal constraints, terminal costs and weights.

If the reference $x^{\mathrm{ref}}$ is time varying, we need to take this fact into account in the formulation of the NMPC algorithm. Similar to the constant case where we assumed that $x_*$ is an equilibrium of (2.1) for control value $u_*$, we now assume that $x^{\mathrm{ref}}$ is a trajectory of the system, i.e.,

$$x^{\mathrm{ref}}(n) = x_{u^{\mathrm{ref}}}(n, x_0)$$

for $x_0 = x^{\mathrm{ref}}(0)$ and some suitable admissible reference control sequence $u^{\mathrm{ref}}(\cdot) \in \mathbb{U}^\infty(x_0)$. In contrast to the constant reference case of Sect. 3.1, even for $X = \mathbb{R}^d$ and $U = \mathbb{R}^m$ we do not assume that these references are constantly equal to 0, because this would lead to time varying coordinate transformations in $X$ and $U$. For this reason, we always need to take $x^{\mathrm{ref}}(\cdot)$ and $u^{\mathrm{ref}}(\cdot)$ into account when defining $\ell$. As a consequence, $\ell$ becomes time varying, too, i.e., we use a function $\ell : \mathbb{N}_0 \times X \times U \to \mathbb{R}_0^+$. Furthermore, we need to keep track of the current sampling instant $n$ in the optimal control problem.

Again, we require that the cost function $\ell$ vanishes if and only if we are exactly on the reference. In the time varying case (3.2) becomes

$$\begin{aligned} &\ell\big(n, x^{\mathrm{ref}}(n), u^{\mathrm{ref}}(n)\big) = 0 \quad \text{for all } n \in \mathbb{N}_0 \quad \text{and} \\ &\ell(n, x, u) > 0 \quad \text{for all } n \in \mathbb{N}_0,\ x \in X,\ u \in U \text{ with } x \neq x^{\mathrm{ref}}(n). \end{aligned} \tag{3.8}$$

For $X = \mathbb{R}^d$, $U = \mathbb{R}^m$ with Euclidean norms, a quadratic distance function is now of the form

$$\ell(n, x, u) = \big\|x - x^{\mathrm{ref}}(n)\big\|^2 + \lambda \big\|u - u^{\mathrm{ref}}(n)\big\|^2$$

with $\lambda \geq 0$ and in the general case

$$\ell(n, x, u) = d_X\big(x, x^{\mathrm{ref}}(n)\big)^2 + \lambda d_U\big(u, u^{\mathrm{ref}}(n)\big)^2$$

is an example for $\ell$ meeting (3.8).

For sampled data systems, we can again define $\ell$ via an integral over a continuous time cost function $L$ analogous to (3.4). Note, however, that for defining $L$ we will then need a continuous time reference.

For each $k = 0, \ldots, N-1$, the prediction $x_u(k, x_0)$ with $x_0 = x(n)$ used in the NMPC algorithm now becomes a prediction for the closed-loop state $x(n+k)$ which we would like to have close to $x^{\mathrm{ref}}(n+k)$. Consequently, in the optimal control problem at time $n$ we need to penalize the distance of $x_u(k, x_0)$ to $x^{\mathrm{ref}}(n+k)$, i.e., we need to use the cost $\ell(n+k, x_u(k, x_0), u(k))$. This leads to the following algorithm where we minimize over the set of control sequences $\mathbb{U}^N(x_0)$ defined in Sect. 3.2.

**Algorithm 3.7** (Basic NMPC algorithm for time varying reference $x^{\mathrm{ref}}$) At each sampling time $t_n$, $n = 0, 1, 2 \ldots$:

(1) Measure the state $x(n) \in X$ of the system.
(2) Set $x_0 = x(n)$, solve the optimal control problem

$$\boxed{\begin{aligned} &\text{minimize} \quad J_N\big(n, x_0, u(\cdot)\big) := \sum_{k=0}^{N-1} \ell\big(n+k, x_u(k, x_0), u(k)\big) \\ &\text{with respect to} \quad u(\cdot) \in \mathbb{U}^N(x_0), \quad \text{subject to} \\ &x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big) \end{aligned}} \tag{$\mathrm{OCP}_N^n$}$$

and denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N(x_0)$.
(3) Define the NMPC-feedback value $\mu_N(n, x(n)) := u^\star(0) \in U$ and use this control value in the next sampling period.

Note that Algorithm 3.7 and ($\mathrm{OCP}_N^n$) reduce to Algorithm 3.1 and ($\mathrm{OCP}_N$), respectively, if $\ell$ does not depend on $n$.

The resulting nominal closed-loop system is now given by (2.5) with $\mu(x) = \mu_N(n, x)$, i.e.,

$$x^+ = f\big(x, \mu_N(n, x)\big). \tag{3.9}$$

As before, the trajectories of this system will be denoted by $x_{\mu_N}(n)$. Since the right hand side is now time varying, whenever necessary we include both the initial time and the initial value in the notation, i.e., for a given $n_0 \in \mathbb{N}_0$ we write $x_{\mu_N}(n, n_0, x_0)$ for the closed-loop solution satisfying $x_{\mu_N}(n_0, n_0, x_0) = x_0$. It is straightforward to check that Theorem 3.5 remains valid for Algorithm 3.7 when (3.7) is replaced by

$$x_{\mu_N}(n) \in \mathbb{X} \quad \text{and} \quad \mu_N\big(n, x_{\mu_N}(n)\big) \in \mathbb{U}\big(x_{\mu_N}(n)\big). \tag{3.10}$$

*Remark 3.8* Observe that Algorithm 3.7 can be straightforwardly extended to the case when $f$ and $\mathbb{X}$ depend on $n$, too. However, in order to keep the presentation simple, we do not explicitly reflect this possibility in our notation.

More often than not one can find variations of the basic NMPC Algorithms 3.1 and 3.7 in the literature in which the optimal control problem ($\mathrm{OCP_N}$) or ($\mathrm{OCP_N^n}$) is changed in one way or another in order to improve the closed-loop performance. These techniques will be discussed in detail in Chap. 5 and in Sects. 7.1 and 7.2. We now introduce generalizations ($\mathrm{OCP_{N,e}}$) and ($\mathrm{OCP_{N,e}^n}$) of ($\mathrm{OCP_N}$) and ($\mathrm{OCP_N^n}$), respectively, which contain all the variants we will investigate in these chapters and sections.

A typical choice for such a variant is an additional terminal constraint of the form

$$x_u\big(N, x(n)\big) \in \mathbb{X}_0 \quad \text{for a } \textit{terminal constraint set } \mathbb{X}_0 \subseteq \mathbb{X} \tag{3.11}$$

for the time-invariant case of ($\mathrm{OCP_N}$) and

$$x_u\big(N, x(n)\big) \in \mathbb{X}_0(n+N) \quad \text{for } \textit{terminal constraint sets } \mathbb{X}_0(n) \subseteq \mathbb{X},\ n \in \mathbb{N}_0 \tag{3.12}$$

for the time varying problem ($\mathrm{OCP_N^n}$). Of course, in the practical implementation the constraint sets $\mathbb{X}_0$ or $\mathbb{X}_0(n)$ are again expressed via (in)equalities of the form given in Definition 3.6.

When using terminal constraints, the NMPC-feedback law is only defined for those states $x_0$ for which the optimization problem within the NMPC algorithm is feasible also for these additional constraints, i.e., for which there exists an admissible control sequence with corresponding trajectory starting in $x_0$ and ending in the terminal constraint set. Such initial values are again called *feasible* and the set of all feasible initial values form the feasible set. This set along with the corresponding admissible control sequences is formally defined as follows.

**Definition 3.9**

(i) For $\mathbb{X}_0$ from (3.11) we define the *feasible set* for horizon $N \in \mathbb{N}$ by

$$\mathbb{X}_N := \big\{x_0 \in \mathbb{X} \,\big|\, \text{there exists } u(\cdot) \in \mathbb{U}^N(x_0) \text{ with } x_u(N, x_0) \in \mathbb{X}_0\big\}$$

and for each $x_0 \in \mathbb{X}_N$ we define the set of *admissible control sequences* by

$$\mathbb{U}_{\mathbb{X}_0}^N(x_0) := \big\{u(\cdot) \in \mathbb{U}^N(x_0) \,\big|\, x_u(N, x_0) \in \mathbb{X}_0\big\}.$$

(ii) For $\mathbb{X}_0(n)$ from (3.12) we define the *feasible set* for horizon $N \in \mathbb{N}$ at time $n \in \mathbb{N}_0$ by

$$\mathbb{X}_N(n) := \big\{x_0 \in \mathbb{X} \,\big|\, \text{there exists } u(\cdot) \in \mathbb{U}^N(x_0) \text{ with } x_u(N, x_0) \in \mathbb{X}_0(n+N)\big\}$$

and for each $x_0 \in \mathbb{X}_N(n)$ we define the set of *admissible control sequences* by

$$\mathbb{U}^N_{\mathbb{X}_0}(n, x_0) := \big\{u(\cdot) \in \mathbb{U}^N(x_0) \,\big|\, x_u(N, x_0) \in \mathbb{X}_0(n+N)\big\}.$$

Note that in (i) $\mathbb{X}_N = \mathbb{X}$ and $\mathbb{U}^N_{\mathbb{X}_0}(x) = \mathbb{U}^N(x)$ holds if $\mathbb{X}_0 = \mathbb{X}$, i.e., if no additional terminal constraints are imposed. Similarly, in case (ii) $\mathbb{X}_N(n) = \mathbb{X}$ and $\mathbb{U}^N_{\mathbb{X}_0}(n, x) = \mathbb{U}^N(x)$ holds if $\mathbb{X}_0(n) = \mathbb{X}$.

Another modification of the optimal control problems ($\mathrm{OCP_N}$) and ($\mathrm{OCP^n_N}$), often used in conjunction with this terminal constraint is an additional terminal cost of the form $F(x_u(N, x(n)))$ with $F : X \to \mathbb{R}^+_0$ in the optimization objective. This function may also be time depending, i.e., it may be of the form $F(n+N, x_u(N, x(n)))$ with $F : \mathbb{N}_0 \times X \to \mathbb{R}^+_0$. An alternative to using terminal costs is to put weights on some summands of the objective, i.e., replacing $\ell(x_u(k, x_0), u(k))$ by $\omega_{N-k}\ell(x_u(k, x_0), u(k))$ for weights $\omega_1, \ldots, \omega_N \geq 0$. Although for NMPC schemes we will only investigate the effect of the weight $\omega_1$ in detail, cf. Sect. 7.2, here we introduce weights for all summands since this offers more flexibility and does not further complicate the subsequent analysis in this chapter. The need for the "backward" numbering of the $\omega_{N-k}$ will become clear in the proof of Theorem 3.15, below.

In the sequel, we will analyze schemes with terminal cost $F$ and schemes with weights $\omega_{N-k}$ separately, cf. Sects. 5.3, 7.1 and 7.2. However, in order to reduce the number of variants of NMPC algorithms in this book we include both features in the optimization problems ($\mathrm{OCP_{N,e}}$) and ($\mathrm{OCP^n_{N,e}}$) in the following NMPC algorithms extending the basic Algorithms 3.1 and 3.7, respectively. Note that compared to these basic algorithms only the optimal control problems are different, i.e., the part in the boxes in Step (2). We start by extending the time-invariant Algorithm 3.1

**Algorithm 3.10** (Extended NMPC algorithm for constant reference $x^{\mathrm{ref}} \equiv x_*$) At each sampling time $t_n$, $n = 0, 1, 2 \ldots$:

(1) Measure the state $x(n) \in X$ of the system.
(2) Set $x_0 := x(n)$, solve the optimal control problem

$$\begin{aligned} &\text{minimize} \quad J_N\big(x_0, u(\cdot)\big) := \sum_{k=0}^{N-1} \omega_{N-k}\ell\big(x_u(k, x_0), u(k)\big) \\ &\qquad\qquad\qquad + F\big(x_u(N, x_0)\big) \\ &\text{with respect to} \quad u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(x_0), \quad \text{subject to} \\ &x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big) \end{aligned} \tag{$\mathrm{OCP_{N,e}}$}$$

and denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(x_0)$.

(3) Define the NMPC-feedback value $\mu_N(x(n)) := u^\star(0) \in U$ and use this control value in the next sampling period.

Similarly, we can extend the time-variant Algorithm 3.7.

**Algorithm 3.11** (Extended NMPC algorithm for time varying reference $x^{\mathrm{ref}}$) At each sampling time $t_n$, $n = 0, 1, 2 \ldots$:

(1) Measure the state $x(n) \in X$ of the system.
(2) Set $x_0 = x(n)$, solve the optimal control problem

$$\begin{aligned} &\text{minimize} \quad J_N\big(n, x_0, u(\cdot)\big) := \sum_{k=0}^{N-1} \omega_{N-k}\, \ell\big(n+k, x_u(k, x_0), u(k)\big) \\ &\qquad\qquad + F\big(n+N, x_u(N, x_0)\big) \\ &\text{with respect to} \quad u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0), \quad \text{subject to} \\ &x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big) \end{aligned} \tag{$\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$}$$

and denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$.
(3) Define the NMPC-feedback value $\mu_N(n, x(n)) := u^\star(0) \in U$ and use this control value in the next sampling period.

Observe that the terminal constraints (3.11) and (3.12) are included via the restrictions $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(x_0)$ and $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$, respectively.

Algorithm 3.10 is a special case of Algorithm 3.11 if $\ell$, $F$ and $\mathbb{X}_0$ do not depend on $n$. Furthermore, Algorithm 3.1 is obtained from Algorithm 3.10 for $F \equiv 0$, $\omega_{N_k} = 1$, $k = 0, \ldots, N-1$ and $\mathbb{X}_0 = \mathbb{X}$. Likewise, we can derive Algorithm 3.7 from Algorithm 3.11 by setting $F \equiv 0$, $\omega_{N_k} = 1$, $k = 0, \ldots, N-1$ and $\mathbb{X}_0(n) = \mathbb{X}$, $n \in \mathbb{N}_0$. Consequently, all NMPC algorithms in this book are special cases of Algorithm 3.11 and all optimal control problems included in these algorithms are special cases of ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$).

We end this section with two useful results on the sets of admissible control sequences from Definition 3.9 which we formulate for the general setting of Algorithm 3.11, i.e., for time varying terminal constraint set $\mathbb{X}_0(n)$.

**Lemma 3.12** *Let $x_0 \in \mathbb{X}_N(n)$, $N \in \mathbb{N}$ and $K \in \{0, \ldots, N\}$ be given.*

(i) *For each $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$ we have $x_u(K, x_0) \in \mathbb{X}_{N-K}(n+K)$.*
(ii) *For each $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$ the control sequences $u_1 \in U^K$ and $u_2 \in U^{N-K}$ uniquely defined by the relation*

$$u(k) = \begin{cases} u_1(k), & k = 0, \ldots, K-1, \\ u_2(k-K), & k = K, \ldots, N-1 \end{cases} \tag{3.13}$$

*satisfy $u_1 \in \mathbb{U}^K_{\mathbb{X}_{N-K}}(n, x_0)$ and $u_2 \in \mathbb{U}^{N-K}_{\mathbb{X}_0}(n+K, x_{u_1}(K, x_0))$.*

(iii) *For each* $u_1(\cdot) \in \mathbb{U}^K_{\mathbb{X}_{N-K}}(n, x_0)$ *there exists* $u_2(\cdot) \in \mathbb{U}^{N-K}_{\mathbb{X}_0}(n+K, x_{u_1}(K, x_0))$ *such that* $u(\cdot)$ *from* (3.13) *satisfies* $u \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$.

*Proof* (i) Using (2.3) we obtain the identity

$$x_{u(K+\cdot)}\big(N-K, x_u(K, x_0)\big) = x_u(N, x_0) \in \mathbb{X}_0(n+N),$$

which together with the definition of $\mathbb{X}_{N-K}$ implies the assertion.

(ii) The relation (3.13) together with (2.3) implies

$$x_u(k, x_0) = \begin{cases} x_{u_1}(k, x_0), & k = 0, \dots, K, \\ x_{u_2}(k-K, x_{u_1}(K, x_0)), & k = K, \dots, N. \end{cases} \tag{3.14}$$

For $k = 0, \dots, K-1$ this identity and (3.13) yield

$$u_1(k) = u(k) \in \mathbb{U}\big(x_u(k, x_0)\big) = \mathbb{U}\big(x_{u_1}(k, x_0)\big)$$

and for $k = 0, \dots, N-K-1$ we obtain

$$u_2(k) = u(k+K) \in \mathbb{U}\big(x_u(k+K, x_0)\big) = \mathbb{U}\big(x_{u_2}\big(k, x_{u_1}(K, x_0)\big)\big),$$

implying $u_1 \in \mathbb{U}^K(x_0)$ and $u_2 \in \mathbb{U}^{N-K}(x_{u_1}(K, x_0))$. Furthermore, (3.14) implies the equation $x_{u_2}(N-K, x_{u_1}(K, x_0)) = x_u(N, x_0) \in \mathbb{X}_0(n+N)$ which proves $u_2 \in \mathbb{U}^{N-K}_{\mathbb{X}_0}(n+K, x_{u_1}(K, x_0))$. This, in turn, implies that $\mathbb{U}^{N-K}_{\mathbb{X}_0}(n+K, x_{u_1}(K, x_0))$ is nonempty, hence $x_{u_1}(K, x_0) \in \mathbb{X}_{N-K}(n+K)$ and consequently $u_1 \in \mathbb{U}^K_{\mathbb{X}_{N-K}}(n, x_0)$ follows.

(iii) By definition, for each $x \in \mathbb{X}_{N-K}(n+K)$ there exists $u_2 \in \mathbb{U}^{N-K}_{\mathbb{X}_0}(n+K, x)$. Choosing such a $u_2$ for $x = x_{u_1}(K, x_0) \in \mathbb{X}_{N-K}(n+K)$ and defining $u$ via (3.13), similar arguments as in Part (ii), above, show the claim $u \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$. □

A straightforward corollary of this lemma is the following.

**Corollary 3.13**

(i) *For each* $x \in \mathbb{X}_N$ *the NMPC-feedback law* $\mu_N$ *obtained from Algorithm* 3.10 *satisfies*

$$f\big(x, \mu_N(x)\big) \in \mathbb{X}_{N-1}.$$

(ii) *For each* $n \in \mathbb{N}$ *and each* $x \in \mathbb{X}_N(n)$ *the NMPC-feedback law* $\mu_N$ *obtained from Algorithm* 3.11 *satisfies*

$$f\big(x, \mu_N(n, x)\big) \in \mathbb{X}_{N-1}(n+1).$$

*Proof* We show (ii) which contains (i) as a special case. Since $\mu_N(n, x)$ is the first element $u^\star(0)$ of the optimal control sequence $u^\star \in \mathbb{U}^N_{\mathbb{X}_0}(n, x)$ we get $f(x, \mu_N(n, x)) = x_{u^\star}(1, x)$. Now Lemma 3.12(i) yields the assertion. □

## 3.4 The Dynamic Programming Principle

In this section we provide one of the classical tools in optimal control, the *dynamic programming principle*. We will formulate and prove the results in this section for ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$), since all other optimal control problems introduced above can be obtained a special cases of this problem. We will first formulate the principle for the open-loop control sequences in ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) and then derive consequences for the NMPC-feedback law $\mu_N$. The dynamic programming principle is often used as a basis for numerical algorithms, cf. Sect. 3.5. In contrast to this, in this book we will exclusively use the principle for analyzing the behavior of NMPC closed-loop systems, while for the actual numerical solution of ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) we use different algorithms as described in Chap. 10. The reason for this is that the numerical effort of solving ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) via dynamic programming usually grows exponentially with the dimension of the state of the system, see the discussion in Sect. 3.5. In contrast to this, the computational effort of the methods described in Chap. 10 scales much more moderately with the space dimension.

We start by defining some objects we need in the sequel.

**Definition 3.14** Consider the optimal control problem ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) with initial value $x_0 \in \mathbb{X}$, time instant $n \in \mathbb{N}_0$ and optimization horizon $N \in \mathbb{N}_0$.

(i) The function

$$V_N(n, x_0) := \inf_{u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(x_0)} J_N\big(n, x_0, u(\cdot)\big)$$

is called *optimal value function*.

(ii) A control sequence $u^\star(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(x_0)$ is called *optimal control sequence* for $x_0$, if

$$V_N(n, x_0) = J_N\big(n, x_0, u^\star(\cdot)\big)$$

holds. The corresponding trajectory $x_{u^\star}(\cdot, x_0)$ is called *optimal trajectory*.

In our NMPC Algorithm 3.11 and its variants we have assumed that an optimal control sequence $u^\star(\cdot)$ exists, cf. the comment after Algorithms 3.1. In general, this is not necessarily the case but under reasonable continuity and compactness conditions the existence of $u^\star(\cdot)$ can be rigorously shown. Examples of such theorems for a general infinite-dimensional state space can be found in Keerthi and Gilbert [10] or Doležal [7]. While for formulating and proving the dynamic programming principle we will not need the existence of $u^\star(\cdot)$, for all subsequent results we will assume that $u^\star(\cdot)$ exists, in particular when we derive properties of the NMPC-feedback law $\mu_N$. While we conjecture that most of the results in this book can be generalized to the case when $\mu_N$ is defined via an approximately minimizing control sequence, we decided to use the existence assumption because it considerably simplifies the presentation of the results in this book.

The following theorem introduces the *dynamic programming principle*. It gives an equation which relates the optimal value functions for different optimization horizons $N$ and for different points in space.

**Theorem 3.15** *Consider the optimal control problem* ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) *with* $x_0 \in \mathbb{X}_N(n)$ *and* $n, N \in \mathbb{N}_0$. *Then for all* $N \in \mathbb{N}$ *and all* $K = 1, \ldots, N$ *the equation*

$$V_N(n, x_0) = \inf_{u(\cdot) \in \mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)} \left\{ \sum_{k=0}^{K-1} \omega_{N-k} \ell\big(n+k, x_u(k, x_0), u(k)\big) + V_{N-K}\big(n+K, x_u(K, x_0)\big) \right\} \tag{3.15}$$

*holds. If, in addition, an optimal control sequence* $u^\star(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$ *exists for* $x_0$, *then we get the equation*

$$V_N(n, x_0) = \sum_{k=0}^{K-1} \omega_{N-k} \ell\big(n+k, x_{u^\star}(k, x_0), u^\star(k)\big) + V_{N-K}\big(n+K, x_{u^\star}(K, x_0)\big). \tag{3.16}$$

*In particular, in this case the* "inf" *in* (3.15) *is a* "min".

*Proof* First observe that from the definition of $J_N$ for $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$ we immediately obtain

$$J_N\big(n, x_0, u(\cdot)\big) = \sum_{k=0}^{K-1} \omega_{N-k} \ell\big(n+k, x_u(k, x_0), u(k)\big) + J_{N-K}\big(n+K, x_u(K, x_0), u(\cdot+K)\big). \tag{3.17}$$

Since $u(\cdot + K)$ equals $u_2(\cdot)$ from Lemma 3.12(ii) we obtain $u(\cdot+K) \in \mathbb{U}^{N-K}_{\mathbb{X}_0}(n + K, x_u(K, x_0))$. Note that for (3.17) to hold we need the backward numbering of $\omega_{N-k}$.

We now prove (3.15) by proving "$\geq$" and "$\leq$" separately. From (3.17) we obtain

$$\begin{aligned} J_N\big(n, x_0, u(\cdot)\big) &= \sum_{k=0}^{K-1} \omega_{N-k} \ell\big(n+k, x_u(k, x_0), u(k)\big) \\ &\quad + J_{N-K}\big(n+K, x_u(K, x_0), u(\cdot+K)\big) \\ &\geq \sum_{k=0}^{K-1} \omega_{N-k} \ell\big(n+k, x_u(k, x_0), u(k)\big) + V_{N-K}\big(n+K, x_u(K, x_0)\big). \end{aligned}$$

Since this inequality holds for all $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n, x_0)$, it also holds when taking the infimum on both sides. Hence we get

$$\begin{aligned} V_N(n, x_0) &= \inf_{u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n,x_0)} J_N\big(n, x_0, u(\cdot)\big) \\ &\geq \inf_{u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n,x_0)} \left\{ \sum_{k=0}^{K-1} \omega_{N-k} \ell\big(n+k, x_u(k, x_0), u(k)\big) \right. \end{aligned}$$

$$
\begin{aligned}
&\quad + V_{N-K}\big(n+K, x_u(K,x_0)\big)\Big\} \\
&= \inf_{u_1(\cdot)\in\mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)} \Big\{ \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_{u_1}(k,x_0), u(k)\big) \\
&\quad + V_{N-K}\big(n+K, x_{u_1}(K,x_0)\big)\Big\},
\end{aligned}
$$

i.e., (3.15) with "$\geq$". Here in the last step we used the fact that by Lemma 3.12(ii) the control sequence $u_1$ consisting of the first $K$ elements of $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n,x_0)$ lies in $\mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)$ and, conversely, by Lemma 3.12(iii) each control sequence in $u_1(\cdot) \in \mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)$ can be extended to a sequence in $u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(n,x_0)$. Thus, since the expression in braces does not depend on $u(K), \ldots, u(N-1)$, the infima coincide.

In order to prove "$\leq$", fix $\varepsilon > 0$ and let $u^\varepsilon(\cdot)$ be an approximately optimal control sequence for the right hand side of (3.17), i.e.,

$$
\begin{aligned}
&\sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_{u^\varepsilon}(k,x_0), u^\varepsilon(k)\big) + J_{N-K}\big(n+K, x_{u^\varepsilon}(K,x_0), u^\varepsilon(\cdot+K)\big) \\
&\quad \leq \inf_{u(\cdot)\in\mathbb{U}^N_{\mathbb{X}_0}(n,x_0)} \Big\{ \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_u(k,x_0), u(k)\big) \\
&\qquad + J_{N-K}\big(n+K, x_u(K,x_0), u(\cdot+K)\big)\Big\} + \varepsilon.
\end{aligned}
$$

Now we use the decomposition (3.13) of $u(\cdot)$ into $u_1 \in \mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)$ and $u_2 \in \mathbb{U}^{N-K}_{\mathbb{X}_0}(n+K, x_{u_1}(K,x_0))$ from Lemma 3.12(ii). This way we obtain

$$
\begin{aligned}
&\inf_{u(\cdot)\in\mathbb{U}^N_{\mathbb{X}_0}(n,x_0)} \Big\{ \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_u(k,x_0), u(k)\big) \\
&\qquad + J_{N-K}\big(n+K, x_u(K,x_0), u(\cdot+K)\big)\Big\} \\
&= \inf_{\substack{u_1(\cdot)\in\mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0) \\ u_2(\cdot)\in\mathbb{U}^{N-K}_{\mathbb{X}_0}(n+K,x_{u_1}(K,x_0))}} \Big\{ \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_{u_1}(k,x_0), u_1(k)\big) \\
&\qquad + J_{N-K}\big(n+K, x_{u_1}(K,x_0), u_2(\cdot)\big)\Big\} \\
&= \inf_{u_1(\cdot)\in\mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)} \Big\{ \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_{u_1}(k,x_0), u_1(k)\big)
\end{aligned}
$$

$$+ V_{N-K}\big(n+K, x_{u_1}(K,x_0)\big)\Big\}.$$

Now (3.17) yields

$$\begin{aligned}
V_N(n,x_0) &\le J_N\big(n,x_0,u^\varepsilon(\cdot)\big)\\
&= \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_{u^\varepsilon}(k,x_0), u^\varepsilon(k)\big)\\
&\quad + J_{N-K}\big(n+K, x_{u^\varepsilon}(K,x_0), u^\varepsilon(\cdot+K)\big)\\
&\le \inf_{u(\cdot)\in\mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)} \Bigg\{ \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_u(k,x_0), u(k)\big)\\
&\quad + V_{N-K}\big(n+K, x_u(K,x_0)\big)\Bigg\} + \varepsilon.
\end{aligned}$$

Since the first and the last term in this inequality chain are independent of $\varepsilon$ and since $\varepsilon > 0$ was arbitrary, this shows (3.15) with "$\le$" and thus (3.15).

In order to prove (3.16) we use (3.17) with $u(\cdot) = u^\star(\cdot)$. This yields

$$\begin{aligned}
V_N(n,x_0) &= J_N\big(n,x_0,u^\star(\cdot)\big)\\
&= \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_{u^\star}(k,x_0), u^\star(k)\big)\\
&\quad + J_{N-K}\big(n+K, x_{u^\star}(K,x_0), u^\star(\cdot+K)\big)\\
&\ge \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_{u^\star}(k,x_0), u^\star(k)\big) + V_{N-K}\big(n+K, x_{u^\star}(K,x_0)\big)\\
&\ge \inf_{u(\cdot)\in\mathbb{U}^K_{\mathbb{X}_{N-K}}(n,x_0)} \Bigg\{ \sum_{k=0}^{K-1} \omega_{N-k}\ell\big(n+k, x_u(k,x_0), u(k)\big)\\
&\quad + V_{N-K}\big(n+K, x_u(K,x_0)\big)\Bigg\}\\
&= V_N(n,x_0),
\end{aligned}$$

where we used the (already proven) Equality (3.15) in the last step. Hence, the two "$\ge$" in this chain are actually "$=$" which implies (3.16). □

The following corollary states an immediate consequence of the dynamic programming principle. It shows that tails of optimal control sequences are again optimal control sequences for suitably adjusted optimization horizon, time instant and initial value.

**Corollary 3.16** *If $u^\star(\cdot)$ is an optimal control sequence for initial value $x_0 \in \mathbb{X}_N(n)$, time instant $n$ and optimization horizon $N \geq 2$, then for each $K = 1, \ldots, N-1$ the sequence $u^\star_K(\cdot) = u^\star(\cdot + K)$, i.e.,*

$$u^\star_K(k) = u^\star(K+k), \quad k = 0, \ldots, N-K-1$$

*is an optimal control sequence for initial value $x_{u^\star}(K, x_0)$, time instant $n+K$ and optimization horizon $N-K$.*

*Proof* Inserting $V_N(n, x_0) = J_N(n, x_0, u^\star(\cdot))$ and the definition of $u^\star_k(\cdot)$ into (3.17) we obtain

$$\begin{aligned} V_N(n, x_0) = &\sum_{k=0}^{K-1} \omega_{N-k}\, \ell\big(n+k, x_{u^\star}(k, x_0), u^\star(k)\big) \\ &+ J_{N-K}\big(n+K, x_{u^\star}(K, x_0), u^\star_K(\cdot)\big). \end{aligned}$$

Subtracting (3.16) from this equation yields

$$0 = J_{N-K}\big(n+K, x_{u^\star}(K, x_0), u^\star_K(\cdot)\big) - V_{N-K}\big(n+K, x_{u^\star}(K, x_0)\big)$$

which shows the assertion. □

The next theorem relates the NMPC-feedback law $\mu_N$ defined in the NMPC Algorithm 3.11 and its variants to the dynamic programming principle. Here we use the argmin operator in the following sense: for a map $a : U \to \mathbb{R}$, a nonempty subset $\widetilde{U} \subseteq U$ and a value $u^\star \in \widetilde{U}$ we write

$$u^\star = \operatorname*{argmin}_{u \in \widetilde{U}} a(u) \tag{3.18}$$

if and only if $a(u^\star) = \inf_{u \in \widetilde{U}} a(u)$ holds. Whenever (3.18) holds the existence of the minimum $\min_{u \in \widetilde{U}} a(u)$ follows. However, we do not require uniqueness of the minimizer $u^\star$. In case of uniqueness equation (3.18) can be understood as an assignment, otherwise it is just a convenient way of writing "$u^\star$ minimizes $a(u)$".

**Theorem 3.17** *Consider the optimal control problem ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) with $x_0 \in \mathbb{X}_N(n)$ and $n, N \in \mathbb{N}_0$ and assume that an optimal control sequence $u^\star$ exists. Then the NMPC-feedback law $\mu_N(n, x_0) = u^*(0)$ satisfies*

$$\mu_N(n, x_0) = \operatorname*{argmin}_{u \in \mathbb{U}^1_{\mathbb{X}_{N-1}}(n, x_0)} \big\{\omega_N \ell(n, x_0, u) + V_{N-1}\big(n+1, f(x_0, u)\big)\big\} \tag{3.19}$$

*and*

$$V_N(n, x_0) = \omega_N \ell\big(n, x_0, \mu_N(n, x_0)\big) + V_{N-1}\big(n+1, f\big(x_0, \mu_N(n, x_0)\big)\big) \tag{3.20}$$

*where in (3.19) we interpret $\mathbb{U}^1_{\mathbb{X}_{N-1}}(n, x_0)$ as a subset of $U$, i.e., we identify the one element sequence $u = u(\cdot)$ with its only element $u = u(0)$.*

*Proof* Equation (3.20) follows by inserting $u^\star(0) = \mu_N(n, x_0)$ and $x_{u^\star}(1, x_0) = f(x_0, \mu_N(n, x_0))$ into (3.16) for $K = 1$.

Inserting $x_u(1, x_0) = f(x_0, u)$ into the dynamic programming principle (3.15) for $K = 1$ we further obtain

$$V_N(n, x_0) = \inf_{u \in \mathbb{U}^1_{\mathbb{X}_{N-1}}(n,x_0)} \big\{ \omega_N \ell(n, x_0, u) + V_{N-1}\big(n+1, f(x_0, u)\big) \big\}. \quad (3.21)$$

This implies that the right hand sides of (3.20) and (3.21) coincide. Thus, the definition of argmin in (3.18) with $a(u) = \omega_N \ell(n, x_0, u) + V_{N-1}(n+1, f(x_0, u))$ and $\widetilde{U} = \mathbb{U}^1_{\mathbb{X}_{N-1}}(n, x_0)$ yields (3.19). □

Our final corollary in this section shows that we can reconstruct the whole optimal control sequence $u^\star(\cdot)$ using the feedback from (3.19).

**Corollary 3.18** *Consider the optimal control problem* ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) *with* $x_0 \in \mathbb{X}$ *and* $n, N \in \mathbb{N}_0$ *and consider admissible feedback laws* $\mu_{N-k} : \mathbb{N}_0 \times \mathbb{X} \to U$, $k = 0, \dots, N-1$, *in the sense of Definition* 3.2(iv). *Denote the solution of the closed-loop system*

$$\begin{aligned} x(0) &= x_0, \\ x(k+1) &= f\big(x(k), \mu_{N-k}\big(n+k, x(k)\big)\big), \quad k = 0, \dots, N-1 \end{aligned} \quad (3.22)$$

*by* $x_\mu(\cdot)$ *and assume that the* $\mu_{N-k}$ *satisfy* (3.19) *with horizon* $N-k$ *instead of* $N$, *time index* $n+k$ *instead of* $n$ *and initial value* $x_0 = x_\mu(k)$ *for* $k = 0, \dots, N-1$. *Then*

$$u^\star(k) = \mu_{N-k}\big(n+k, x_\mu(k)\big), \quad k = 0, \dots, N-1 \quad (3.23)$$

*is an optimal control sequence for initial time* $n$ *and initial value* $x_0$ *and the solution of the closed-loop system* (3.22) *is a corresponding optimal trajectory.*

*Proof* Applying the control (3.23) to the dynamics (3.22) we immediately obtain

$$x_{u^\star}(n, x_0) = x_\mu(n), \quad n = 0, \dots, N-1.$$

Hence, we need to show that

$$V_N(n, x_0) = J_N\big(n, x_0, u^\star\big) = \sum_{k=0}^{N-1} \omega_{N-k} \ell\big(n+k, x(k), u^\star(k)\big) + F\big(n+N, x(N)\big).$$

Using (3.23) and (3.20) for $N-k$ instead of $N$ we get

$$V_{N-k}(n+k, x_0) = \omega_{N-k} \ell\big(n+k, x(k), u^\star(k)\big) + V_{N-k-1}\big(n+k+1, x(k+1)\big)$$

for $k = 0, \dots, N-1$. Summing these equalities for $k = 0, \dots, N-1$ and eliminating the identical terms $V_{N-k}(n+k, x_0)$, $k = 1, \dots, N-1$ on both sides we obtain

$$V_N(n, x_0) = \sum_{k=0}^{N-1} \omega_{N-k} \ell\big(n+k, x(k), u^\star(k)\big) + V_0\big(n+N, x(N)\big).$$

Since by definition of $J_0$ we have $V_0(n+N, x) = F(n+N, x)$, this shows the assertion. □

## 3.5 Notes and Extensions

The discrete time nonlinear model predictive control framework introduced in Sects. 3.1–3.3 covers most of the settings commonly found in the literature. For continuous time systems, one often also finds nonlinear model predictive control frameworks in explicit continuous time form. In these frameworks, the optimization in ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) and its variants is carried out at times $t_0, t_1, t_2, \ldots$ minimizing an integral criterion along the continuous time solution of the form

$$J_{T_{\mathrm{opt}}}(x_0, v) = \int_0^{T_{\mathrm{opt}}} L\big(\varphi(t, x_0, v), v(t)\big)\, dt + F\big(\varphi(T_{\mathrm{opt}}, N, x_0, v)\big).$$

The feedback law $\mu_{T_{\mathrm{opt}}}$ computed at time $t_n$ is then obtained by applying the first portion $v^\star|_{[0,t_{n+1}-t_n]}$ of the optimal control function $v^\star$ to the system, see, e.g., Alamir [1] or Findeisen [9]. Provided that $t_{n+1} - t_n = T$ holds for all $n$, this problem is equivalent to our setting if the sampled data system (2.8) and the integral criterion (3.4) is used.

Regarding notation, in NMPC it is important to distinguish between the open-loop predictions and the NMPC closed loop. Here we have decided to denote the open-loop predictions by $x_u(k)$ or $x_u(k, x_0)$ and the NMPC closed-loop trajectories by either $x(n)$ or—more often—by $x_{\mu_N}(n)$ or $x_{\mu_N}(n, x_0)$. There are, however, various other notations commonly found in the literature. For instance, the prediction at time instant $n$ is occasionally denoted as $x(k|n)$ in order to emphasize the dependence on the time instant $n$. In our notation, the dependence on $n$ is implicitly expressed via the initial condition $x_0 = x(n)$ and the index $n$ in ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N}}$) or ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$). Whenever necessary, the value of $n$ under consideration will be specified in the context. On the other hand, we decided to always explicitly indicate the dependence of open-loop solutions on the control sequence $u$. This notation enables us to easily distinguish between open-loop and closed-loop solutions and also for simultaneously considering open-loop solutions for different control sequences.

In linear discrete time MPC, the optimization at each sampling instant is occasionally performed over control sequences with predefined values $u(K), \ldots, u(N-1)$ for some $K \in \{1, \ldots, N-1\}$, i.e., only $u(0), \ldots, u(K-1)$ are used as optimization variables in ($\mathrm{OCP}_{\mathrm{N,e}}$) and its variants. For instance, if $x_* = 0$ and $u_* = 0$, cf. Sect. 3.1, then $u(K), \ldots, u(N-1) = 0$ is a typical choice. In this setting, $K$ is referred to as *optimization horizon* (or *control horizon*) and $N$ is referred to as *prediction horizon*. Since this variant is less common in nonlinear MPC, we do not consider it in this book; in particular, we use the terms optimization horizon and prediction horizon synonymously, while the term control horizon will receive a different meaning in Sect. 7.4. Still, most of the subsequent analysis could be extended to the case in which the optimization horizon and the prediction horizon do not coincide.

Regarding the cost function $\ell$, the setting described in Sects. 3.1 and 3.3 is easily extended to the case in which a set instead of a single equilibrium or a time-variant family of sets instead of a single reference shall be stabilized. Indeed, if we are given

a family of sets $X^{\mathrm{ref}}(n) \subset X$ such that for each $x \in X^{\mathrm{ref}}(n)$ there is a control $u_x$ with $f(x, u_x) \in X^{\mathrm{ref}}(n+1)$, then we can modify (3.8) to

$$\begin{aligned} &\ell(n, x, u_x) = 0 \quad \text{for all } x \in X^{\mathrm{ref}}(n) \quad \text{and} \\ &\ell(n, x, u) > 0 \quad \text{for all } x \in X \setminus X^{\mathrm{ref}}(n),\ u \in U. \end{aligned} \tag{3.24}$$

Similarly, we can modify (3.2) in the time-invariant case.

Another modification of $\ell$, again often found in the linear MPC literature, are running cost functions which include two consecutive control values, i.e., $\ell(x_u(k), u(k), u(k-1))$. Typically, this is used in order to penalize large changes in the control input by adding a term $\sigma\|u(k) - u(k-1)\|$ (assuming $U$ to be a vector space with norm $\|\cdot\|$, for simplicity). Using the augmented state $\tilde{x}_u(k) = (x_u(k), u(k-1))$ this can be transformed into a cost function meeting our setting by defining $\tilde{\ell}(\tilde{x}_u(k), u(k)) = \ell(x_u(k), u(k), u(k-1))$.

Yet another commonly used variant are running costs in which only an output $y = h(x)$ instead of the whole state is taken into account. In this case, $\ell$ will usually no longer satisfy (3.2) or (3.8), i.e., $\ell$ will not be positive definite, anymore. We will discuss this case in Sect. 7.3. In this context it should be noted that even if the running cost $\ell$ depends only on an output, the NMPC-feedback $\mu_N$ will nevertheless be a state feedback law. Hence, if only output data is available, suitable observers need to be used in order to reconstruct the state of the system.

The term dynamic programming was introduced by Bellman [2] and due to his seminal contributions to this area the dynamic programming principle is often also called Bellman's principle of optimality. The principle is widely used in many application areas and a quite comprehensive account of its use in various different settings is given in the monographs by Bertsekas [4, 5]. For $K = 1$, the dynamic programming principle (3.15) simplifies to

$$V_N(n, x) = \inf_{u \in \mathbb{U}^1_{\mathbb{X}_{N-1}}(n,x)} \big\{\omega_N \ell(n+k, x, u) + V_{N-1}\big(n+1, f(x, u)\big)\big\} \tag{3.25}$$

and in this form it can be used for recursively computing $V_1, V_2, \ldots, V_N$ starting from $V_0(n, x) = F(n, x)$. Once $V_N$ and $V_{N-1}$ are known, the feedback law $\mu_N$ can be obtained from (3.19).

Whenever $V_N$ can be expressed using simple functions this approach of computing $V_N$ can be efficiently used. For instance, when the dynamics are linear and finite dimensional, the running cost is quadratic and there are no constraints, then $V_N$ can be expressed as $V_N(x) = x^\top P_N x$ for a matrix $P_N \in \mathbb{R}^{d \times d}$ and (3.25) reduces to the Riccati difference equation, see, e.g., Dorato and Levis [8].

For nonlinear systems with low-dimensional state space it is also possible to approximate $V_N$ numerically using the backward recursion induced by (3.25) with approximations $\widetilde{V}_1 \approx V_1, \ldots, \widetilde{V}_N \approx V_N$. These approximations can then, in turn, be used in order to compute a numerical approximation of the NMPC-feedback law $\mu_N$. This is, roughly speaking, the idea behind the so-called *explicit MPC* methods, see, e.g., Borrelli, Baotic, Bemporad and Morari [6], Bemporad and Filippi [3], Tøndel, Johansen and Bemporad [11], to mention but a few papers from this area in which often special problem structures like piecewise linear dynamics instead of

general nonlinear models are considered. The main advantage of this approach is that $\widetilde{V}_N$ and the approximation of $\mu_N$ can be computed offline and thus the online computational effort of evaluating $\mu_N$ is very low. Hence, in contrast to conventional NMPC in which ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) is entirely solved online, this method is also applicable to very fast systems which require fast sampling.

Unfortunately, for high-dimensional systems, the numerical effort of this approach becomes prohibitive since the computational complexity of computing $\widetilde{V}_N$ grows exponentially in the state dimension, unless one can exploit very specific problem structure. This fact—the so-called curse of dimensionality—arises because the approximation of $V_N$ requires a global solution to ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) or its variants for all initial values $x_0 \in \mathbb{X}$ or at least in the set of interest, which is typically a set of full dimension in state space. Consequently, the dynamic programming method cannot be applied to high-dimensional systems. In contrast to this, the methods we will discuss in Chap. 10 solve ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}}$) for a single initial value $x_0$ only at each sampling instant, i.e. locally in space. Since this needs to be done online, these methods are in principle slower, but since the numerical effort scales much more moderate with the state dimension they are nevertheless applicable to systems with much higher state dimension.

## 3.6 Problems

1. Consider the control system

$$x^+ = f(x,u) = ax + bu$$

with $x \in X = \mathbb{R}$, $u \in U = \mathbb{R}$, constraints $\mathbb{X} = [-1, 1]$ and $\mathbb{U} = [-100, 100]$ and real parameters $a, b \in \mathbb{R}$.

   (a) For which parameters $a, b \in \mathbb{R}$ is the state constraint set $\mathbb{X}$ viable?

   (b) For those parameters for which $\mathbb{X}$ is not viable, determine a viable state constraint set contained in $\mathbb{X}$.

2. Compute an optimal trajectory for the optimal control problem ($\mathrm{OCP}_{\mathrm{N,e}}$)

$$\begin{aligned} \text{minimize} \quad & \sum_{k=0}^{N-1} u(k)^2, \\ \text{subject to} \quad & x_1(k+1) = x_1(k) + 2x_2(k), \\ & x_2(k+1) = 2u(k) - x_2(k), \\ & x_1(0) = 0, \qquad x_2(0) = 0, \\ & x_1(N) = 4, \qquad x_2(N) = 0 \end{aligned}$$

with $N = 4$ via dynamic programming.

3. Consider the NMPC problem defined by the dynamics

$$x^+ = f(x,u) = x + u$$

with $x \in X = \mathbb{R}$, $u \in U = \mathbb{R}$ and running costs

$$\ell(x, u) = x^2 + u^2.$$

(a) Compute the optimal value function $V_2$ and the NMPC-feedback law $\mu_2$ by dynamic programming.
(b) Show that $V_2$ is a Lyapunov function for the closed loop and compute the functions $\alpha_1$, $\alpha_2$ and $\alpha_V$ in (2.37) and (2.38).
(c) Show that the NMPC closed loop is globally asymptotically stable without using the Lyapunov function $V_2$.

4. Consider an optimal trajectory $x_{u^\star}(\cdot, x_0)$ for the optimal control problem ($\mathrm{OCP_N}$) with initial value $x_0$ and optimization horizon $N \geq 2$. Prove that for any $K \in \{1, \ldots, N-1\}$ the tail

$$x_{u^\star}(K, x_0), \ldots, x_{u^\star}(N-1, x_0)$$

of the optimal trajectory along with the tail

$$u^\star(K), \ldots, u^\star(N-1)$$

of the optimal control sequence are optimal for ($\mathrm{OCP_N}$) with new initial value $x_{u^\star}(K, x_0)$ and optimization horizon $N - K$, i.e., that

$$\sum_{k=K}^{N-1} \ell\big(x_{u^\star}(k, x_0), u^\star(k)\big) = V_{N-K}\big(x_{u^\star}(K, x_0)\big)$$

holds.

5. After a lecture in which you presented the basic NMPC Algorithm 3.1, a student asks the following question:

"If I ride my bicycle and want to make a turn to the left, I first steer a little bit to the right to make my bicycle tilt to the left. Let us assume that this way of making a turn is optimal for a suitable problem of type ($\mathrm{OCP_N}$). This would mean that the optimal control sequence will initially steer to the right and later steer to the left. If we use this optimal control sequence in an NMPC algorithm, only the first control action will be implemented. As a consequence, we will always steer to the right, and we will make a turn to the right instead of a turn to the left. Does this mean that NMPC does not work for controlling my bicycle?"

What do you respond?

## References

1. Alamir, M.: Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes. Lecture Notes in Control and Information Sciences, vol. 339. Springer, London (2006)
2. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton (1957). Reprinted in 2010
3. Bemporad, A., Filippi, C.: Suboptimal explicit MPC via approximate multiparametric quadratic programming. In: Proceedings of the 40th IEEE Conference on Decision and Control – CDC 2001, Orlando, Florida, USA, pp. 4851–4856 (2001)

4. Bertsekas, D.P.: Dynamic Programming and Optimal Control, vol. I, 3rd edn. Athena Scientific, Belmont (2005)
5. Bertsekas, D.P.: Dynamic Programming and Optimal Control, vol. II, 2nd edn. Athena Scientific, Belmont (2001)
6. Borrelli, L., Baotic, T., Bemporad, A., Morari, T.: Efficient on-line computation of constrained optimal control. In: Proceedings of the 40th IEEE Conference on Decision and Control – CDC 2001, Orlando, Florida, USA, pp. 1187–1192 (2001)
7. Doležal, J.: Existence of optimal solutions in general discrete systems. Kybernetika **11**(4), 301–312 (1975)
8. Dorato, P., Levis, A.H.: Optimal linear regulators: the discrete-time case. IEEE Trans. Automat. Control **16**, 613–620 (1971)
9. Findeisen, R.: Nonlinear model predictive control: a sampled-data feedback perspective. PhD thesis, University of Stuttgart, VDI-Verlag, Düsseldorf (2004)
10. Keerthi, S.S., Gilbert, E.G.: An existence theorem for discrete-time infinite-horizon optimal control problems. IEEE Trans. Automat. Control **30**(9), 907–909 (1985)
11. Tøndel, P., Johansen, T.A., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit MPC solutions. Automatica **39**(3), 489–497 (2003)

http://www.springer.com/978-0-85729-500-2

Nonlinear Model Predictive Control
Theory and Algorithms
Grüne, L.; Pannek, J.
2011, XII, 360 p. With online files/update., Hardcover
ISBN: 978-0-85729-500-2